Please amend the claims as follows. This Listing of Claims will replace all prior versions, and Listings of Claims in the application:

Listing of Claims:

1. (Currently amended) A computer system comprising:

a processor;

a memory storage unit;

an operating system comprising a kernel, said kernel comprising a plurality of kernel modules, said kernel modules comprising signature information; and

a kernel module signature verification system for verifying said kernel module signature information of each of said plurality of kernel modules as said plurality of kernel modules are loaded into said kernel,

wherein said kernel module signature information is generated via a public key and a private key compilation in said kernel module.

2. (Cancel)

3. (Currently amended) The computer system of claim 1 [[2]], wherein said kernel module signature information comprises signature length data unique to each of said plurality of kernel modules, said signature length data used by said kernel module signature verification system in uniquely identifying each of said plurality of kernel modules.

4. (Original) The computer system of claim 3, wherein said kernel module signature information further comprises signature size data for further uniquely identifying each of said kernel module.

5. (Original) The computer system of claim 4, wherein said kernel module signature verification system comprises a kernel cryptographic framework for verifying said kernel module signature information.

6. (Original) The computer system of claim 5, wherein said kernel module signature verification system further comprises a kernel cryptographic framework daemon for performing verification lookup operations of signature information provided to said kernel cryptographic framework in said kernel.

7. (Original) The computer system of claim 6, wherein said kernel cryptographic framework daemon further performs module verification of said plurality of kernel modules.

8. (Original) The computer system of claim 7, wherein said kernel cryptographic framework retrieves pathname information of said signature information for each of said plurality of kernel modules when said plurality of kernel modules attempt to load up to said kernel to perform cryptographic operations.

9. (Original) The computer system of claim 8, wherein said kernel cryptographic framework comprises a cryptographic service provider registration unit for registering each of said plurality of kernel modules wishing to provide cryptographic services in said kernel.

10. (Original) The computer system of claim 9, wherein said kernel cryptographic framework further comprises a intra-kernel communication unit for enabling communications between said kernel cryptographic framework and said kernel cryptographic framework daemon.

11. (Original) The computer system of claim 10, wherein said kernel cryptographic

framework further comprises a data structure unit for storing said kernel module signature information.

12. (Currently amended) A computer operating system comprising:

a memory storage unit;

a kernel, said kernel comprising a plurality of kernel modules; and

a kernel module signature verification system for verifying signature information of said plurality of kernel modules as said plurality of kernel modules are loaded into said kernel;

wherein said kernel signature information further comprises signature length data for further uniquely identifying each one of said plurality of kernel modules.

13. (Original) The computer operating system of claim 12, wherein said kernel signature information comprises kernel signature data for uniquely identifying each one of said plurality of kernel modules.

14. (Cancel)

15. (Currently amended) The computer operating system of claim 12 [[14]], wherein said kernel signature information further comprises signature size data for each of said plurality of kernel modules.

16. (Original) The computer operating system of claim 15, wherein said kernel module signature verification system comprises a kernel cryptographic framework for authorizing and verifying signature information of kernel cryptographic modules loading into said kernel

to provide kernel cryptographic services.

17. (Original) The computer operating system of claim 16, wherein said kernel module signature verification system further comprises a kernel cryptographic framework daemon.

18. (Original) The computer operating system of claim 17, wherein said kernel cryptographic framework daemon performs module verification of said plurality of kernel modules.

19. (Original) The computer operating system of claim 18, wherein said kernel cryptographic framework retrieves pathname information of said signature information for each of said plurality of kernel modules when said plurality of kernel modules attempt to load up to said kernel to perform cryptographic operations.

20. (Original) The computer operating system of claim 19, wherein said kernel cryptographic framework comprises a cryptographic service provider registration unit for registering each of said plurality of kernel modules wishing to provide cryptographic services in said kernel.

21. (Original) The computer operating system of claim 20, wherein said kernel cryptographic framework further comprises an intra-kernel communication unit for enabling communications between said kernel cryptographic framework and said kernel cryptographic framework daemon.

22. (Original) The computer operating system of claim 21, wherein said kernel cryptographic framework further comprises a data structure unit for storing said kernel module signature information.

23. (Original) The computer operating system of claim 22, wherein said kernel cryptographic framework and said kernel cryptographic framework daemon communicate via a plurality of input/output control commands.

24. (Original) The computer operating system of claim 23, wherein said input/output control commands comprise a door create command for creating a plurality of cryptographic doors for enabling communication between said kernel cryptographic framework and said kernel cryptographic framework daemon.

25. (Currently amended) In a computer system, a computer software implemented kernel module signature verification system, comprising:

kernel cryptographic framework for verifying signatures uniquely defining each of a plurality of kernel cryptographic modules; and

kernel cryptographic framework daemon for performing module verification for each of said plurality of kernel cryptographic modules, wherein said kernel cryptographic framework daemon retrieves pathname information of said signature information for each of said plurality of kernel modules when said plurality of kernel modules attempt to load up to said kernel to perform cryptographic operations.

26. (Cancel)

27. (Currently amended) The kernel module signature verification system of claim 25 [[26]], wherein said kernel cryptographic framework comprises a cryptographic service provider registration unit for registering each of said plurality of kernel modules wishing to provide cryptographic services in said kernel.

28. (Original) The kernel module signature verification system of claim 27, wherein said kernel cryptographic framework further comprises an intra-kernel communication unit for enabling communications between said kernel cryptographic framework and said kernel cryptographic framework daemon.

29. (Original) The kernel module signature verification system of claim 28, wherein said kernel cryptographic framework further comprises a data structure unit for storing said kernel module signature information.

30. (Original) The kernel module signature verification system of claim 29, wherein said kernel cryptographic framework and said kernel cryptographic framework daemon communicate via a plurality of input/output control commands.

31. (Currently amended) A method of verifying and authenticating kernel cryptographic modules, said method comprising:

providing a kernel cryptographic framework for verifying signature data in each of a plurality of kernel cryptographic modules, said kernel cryptographic framework accepts registration requests from a requesting kernel module of said plurality of kernel cryptographic modules to register and load as cryptographic service providers and said kernel cryptographic framework verifies whether said results from verifying said signature data of said requesting kernel module compares with signature information stored in said kernel cryptographic framework to authenticate said requesting kernel module; and

providing a kernel cryptographic framework daemon for communicating with said kernel cryptographic framework for performing module verification of said plurality of kernel cryptographic modules.

32. (Original) The method of claim 31, wherein said kernel cryptographic framework daemon creates an unnamed door that is passed to establish communication between said kernel cryptographic framework and said kernel cryptographic framework daemon.

33. (Cancel)

34. (Currently amended) The method of claim 31 [[33]], wherein said kernel cryptographic framework daemon verifies signature data contained in each of said plurality of kernel cryptographic modules after said requesting kernel module has registered with said kernel cryptographic framework.

35. (Original) The method of claim 34, wherein said kernel cryptographic framework daemon passes results from verifying said signature data of said requesting kernel module to said kernel cryptographic framework.

36. (Cancel)